

# Speed Control of Induction Motor by Z-N Method and Genetic Algorithm Optimization with PI and PID Controller

Jaya Raju Manepalli<sup>1</sup>, CH.V.N. Raja<sup>2</sup>

PG Student, Department of EEE, ANITS, Visakhapatnam, Andhra Pradesh, India<sup>1</sup>

Associate Professor, Department of EEE, ANITS, Visakhapatnam, Andhra Pradesh, India<sup>2</sup>

**Abstract:** This Paper presents a comparative study of Z-N method and Genetic Algorithm method (GA) to determine the optimal proportional-integral-derivative (PID) controller parameters, for speed control of a Field Oriented Control (FOC) induction motor; the GA algorithm has been programmed and implemented in MATLAB. Z-N method and trial and error and open loop IM has been modelled in MATLAB (SIMULINK). comparing with traditional Ziegler-Nicholson method, it has been observed that during optimizing the controller parameters of a FOC IM drive with evolutionary algorithms (EA), the performance of the controller is improved for the step input in speed control as well as for speed tracking problem more efficiently under no load condition, if the load is placed on IM, the performance characteristics have changed for ZN and trial and error method, but even if load change occur, there is no much variation in the evolutionary algorithms (GA) than and Ziegler – Nicholson method.

**Keywords:** FOC (IM), fitness function, GA, Z-N method, PIControl, PID control

## I. INTRODUCTION

The induction motors has been widely used in various industries due to its robustness maintenance free operation, better efficiency and lower cost. In different industries, wide range of speed control with fast torque response regardless of load variation is required this can be achieved very efficiently for induction motor using Field Oriented Control (FOC) [1, 2]. For speed control of induction motor, PI (proportional-integral) and PID (Proportional-integral-derivative) controllers are generally used. To find out the optimum parameters of the controller to obtain a good closed loop response at different operating conditions is a trivial task and these parameters can be optimized by conventional tuning methods, such as Ziegler-Nicholson (Z-N) method [6]. Other tuning methods like pole placement optimization technique are also done [4]. Now a day, Evolutionary methods like Genetic Algorithm (GA), is used for tuning the parameters. These new tuning techniques can very efficiently solve complex problems like speed tracking problems, where demand speed is a complex function of time, where the conventional methods may not optimize the controller parameter so easily. Genetic Algorithm is a heuristics search method based on Charles's Darwin principle of Natural Selection which narrates 'the survival of the fittest' of each and every individual on earth. At each step, the GA selects individuals from the current population as parents and uses them to produce the offspring's for the next generations. The fitness of all the individual of the population is calculated and the convergence of the generation is based on this fitness criterion. It is well suited for its solving complex design Optimization problem as it can handle discrete and continuous variables, nonlinearity and different constrain functions of a system, without requiring gradient information [9].

The major objective of this work is to compare efficiency of both Z-N method and GA optimization technique Applied to a direct field oriented Control Induction motor drive for a simple speed demand problem as well as for a complex speed problem.

Here both Z-N method and GA have been applied to search for the optimal PID controller parameters of FOC IM drive. The error criteria for both the methods are set to improve transient error and steady state error

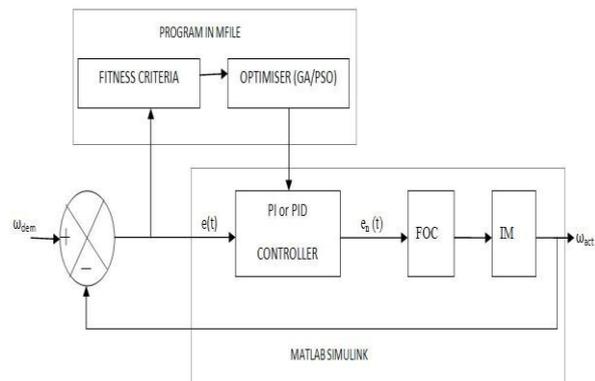


Fig. 1: Schematic diagram of GA based optimizer for PI and PID controller of FOC IM

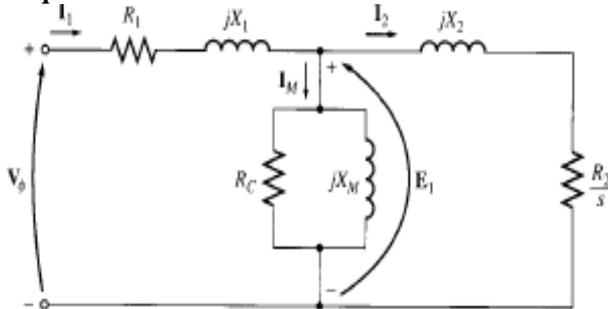
The PID controller's gain parameters viz.  $K_p$ ,  $K_i$  and  $K_d$  are optimized, by GA, to have the optimum output of the controllers are given by Eqn.4 and Eqn.5 [6, 7, 8].

Here  $e(t)$  is the difference between the demand speed and the actual speed of the system is denoted by  $\omega_{dem}$  and  $\omega_{act}$ . For the speed tracking problem, the parameters are optimized obeying the same procedure as stated above.

## II. MATHEMATICAL MODELLING OF INDUCTION MOTOR

The voltage and torque equations that describe the dynamic behaviour of an induction motor are time varying. Differential equations involve some complexity. A change of variables can be used to reduce the complexity of these equations by eliminating all time-varying inductances. By this approach, a poly phase winding can be reduced to a set of two phase windings (q-d) with their magnetic axis formed in quadrature. In other words, the stator and rotor variables (voltages, currents and flux linkages) of an induction machine are transferred to a reference frame, which may rotate at any angular velocity or remain stationary. Such a frame of reference is commonly known in the generalized machines analysis as arbitrary reference frame. The dynamic analysis of the symmetrical induction machines in the arbitrary reference frame has been intensively used as a standard simulation approach from which any particular mode of operation may then be developed. It can be a powerful technique in implementing the machine equations as they are transferred to a particular reference frame. Thus, every single equation among the model equations can be easily implemented in one block so that all the machine variables can be made available for control and verification purposes.

### A. Equivalent circuit of induction motor



The sum of the stator leakage inductance and magnetizing inductance is called the stator inductance ( $L_s = L_{ls} + L_m$ )

And the sum of the rotor leakage inductance and magnetizing inductance is called the rotor inductance ( $L_r = L_{lr} + L_m$ )

Where we have the following equations

$$L_s = \frac{X_s}{\omega}$$

$$L_r = \frac{X_r}{\omega}$$

The flux linkages can be achieved as follows

$$\begin{aligned}\phi_{qs} &= L_s i_{qs} + L_m i_{qr} \\ \phi_{ds} &= L_m i_{dr} + L_s i_{ds} \\ \phi_{qr} &= L_r i_{qr} + L_m i_{qs} \\ \phi_{dr} &= L_r i_{dr} + L_m i_{ds}\end{aligned}$$

The voltage equations are as following:

$$\begin{aligned}v_{qs} &= R_s i_{qs} + \frac{d\phi_{qs}}{dt} + \omega \phi_{ds} \\ v_{ds} &= R_s i_{ds} + \frac{d\phi_{ds}}{dt} - \omega \phi_{qs} \\ v_{qr} &= R_r i_{qr} + \frac{d\phi_{qr}}{dt} + (\omega - \omega_r) \phi_{ds} \\ v_{dr} &= R_r i_{dr} + \frac{d\phi_{dr}}{dt} + (\omega - \omega_r) \phi_{qs}\end{aligned}$$

The torque equation is:

$$T_e = \frac{3}{2} \frac{p}{2} \frac{1}{\omega_b} (\phi_{ds} i_{qs} - \phi_{qs} i_{ds})$$

According to the single phase circuit of the induction motor one can write current equations of stator and rotor in the d-q axis as follows:

$$\begin{aligned}i_{qs} &= \frac{\phi_{qs} - \phi_{mq}}{X_{ls}} \\ i_{ds} &= \frac{\phi_{ds} - \phi_{md}}{X_{ls}} \\ i_{qr} &= \frac{\phi_{qr} - \phi_{mq}}{X_{ls}} \\ i_{dr} &= \frac{\phi_{dr} - \phi_{md}}{X_{ls}}\end{aligned}$$

The speed  $\omega_r$  in the above equations is related to the torque by the following mechanical dynamic equation

$$T_e = T_{load} + j \frac{d\omega_m}{dt} = T_{load} + \frac{2j}{p} \frac{d\omega_r}{dt}$$

Then we can rewrite the above equation for  $\omega_r$  as follows

$$\omega_r = \int \frac{p}{2j} (T_e - T_l)$$

### B. Field Oriented Control:

By Field Oriented Control (FOC) the transient response of IM improves since IM can be controlled like dc machine where its torque component and field flux component are separated virtually and independent control of each component is possible [1,3,8]. FOC is based on phase transformation (park and Clark), where a three phase time and space variant system is transformed to a synchronously rotating, time invariant system, leading to a structure similar to that of a dc machine [1, 2, 3, 4, 5,14]. Thus by this transformation, the three phase time varying system is transformed into two phase time-invariant system, where  $i_{qe}$  corresponds to the Armature component and  $i_{de}$  corresponds to the flux component like a dc machine. Hence the torque ( $T_e$ ) of an IM can be computed as shown in above. Where  $\psi_r$  is the peak value of the field flux space vector. In FOC control here  $i_{ds}$  is analogous to the field current,  $i_f$  and  $i_{qe}$  is analogous to armature current of the dc machine [1].

This means that when  $i_{qe}$  is controlled it affects the torque directly,  $\psi_r$  remaining unaffected. Similarly when  $i_{ds}$  is controlled it affects the flux only and does not affect the  $i_{qs}$  component of current. Thus an induction motor can be treated as a dc machine [1,8,14]. Independent direct torque control can be possible by only controlling the q-axis current control [1,3,4,8]

**C. PI/PID controller:**

The PID controller is a generic control loop feedback mechanism (controller) widely used in industrial control systems [6, 8, 15]. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs (6, 7, 8, 15). The PID controller calculation involves three separate parameters; the proportional, the integral and derivative values, and is given by

$$r(t) = K_p e(t) + K_i \int e(t) dt + K_d de(t)$$

Where  $K_p$ ,  $K_i$ ,  $K_d$  are the proportional, integral and derivative gain of the system.  $u(t)$  is the input signal and  $e(t)$  is the error signal. For PI controller calculation involves two parameters, proportional and integral values and is given by

$$r(t) = K_p e(t) + K_i \int e(t) dt$$

**D. Optimization of PI/PID controller using Z-N method:**

From all the methods designed to optimize PID controller, Ziegler and Nichols' method is mostly used [8,14,15]. The methods are based on characterization of process dynamics by a few parameters and simple equations for the controller parameters.

The first method is applied to plants with step responses [8,14,15]. This type of response is typical of a first order system with transportation delays. The second method targets plants that can be rendered unstable under proportional control.

The technique is designed to result in a closed loop system with 25% overshoot [8,14,15]. This is rarely achieved as Ziegler and Nichols determined the adjustments based on a specific plant model.

Here the second methods have been used,  $K_{cr}$  is the gain at critical oscillation and  $P_{cr}$  is the time period [8,14,15]. The controller gains are specified in the table no: 1

Table1:PID Controller Gain calculation by z-n method

PID type	$K_p$	$K_i$	$K_d$
P	$0.5K_{cr}$	$\infty$	0
PI	$0.45K_{cr}$	$P_{cr}/1.2$	0
PID	$0.6K_{cr}$	$P_{cr}/2$	$P_{cr}/8$

**E. Genetic Algorithm:**

Evolutionary Algorithm (EA) has been employed for solving optimization problems quite successfully. Instead of minimizing or maximizing the object function, it starts with an initial set of parameter values selected randomly. The objective function is evaluated with these parameters and those sets for which the value of the objective function is lower (or higher, as the need may be) are retained and other sets are manipulated. Thus a new set of parameters is evolved from the initial ones and the process is repeated until a "best" choice of parameters is obtained for which the objective function is minimum (or maximum)[8,15].

Among the various evolutionary approaches, Genetic Algorithm can effectively tackle the optimization problem. It is characterized by the chromosome representation, population size, crossover and mutation, their probability rate settings, selection mechanism and fitness function [8, 9,15].

GA requires encoding the solution of an optimization problem in the form of binary strings. The coded parameter in a string represents the chromosome of a particular individual in a population. A large population size incorporates more variation, i.e., diversity into the population, but the convergence becomes slower [8, 9,15].

This ensures the possibility of producing individuals with better fitness. Two randomized methods are incorporated in the algorithm for producing future generations, crossover and mutation. In crossover a partial exchange of genes occur between two parent chromosomes [8,9,15]. The simplest way to achieve this is single point crossover where, a random location of chromosome is selected. If probability of a particular bit in a chromosome exceeds of a pre specified probability i.e.,  $p > p_{cross}$  (defined), the portion of the chromosome of one of the parent, preceding the selected point is combined with the portion of other parent, following the selected bit. In mutation, the parental characteristics are transferred with a slight change in gene[8,9,15]. A selected cell at random is toggled if the probability exceeds the probability of mutation, i.e. if  $p > p_{mut}$ . The choice of the probability rate for crossover and mutation is very much dependent on the population size [8,9,15].

The fitness function provides information regarding the goodness of a particular individual of a generation. Fitness functions generally consist of physical equations representing the objective function that is subject to

optimization. In a GA, the fitter chromosomes are allowed to reproduce with higher probability and thus propagate into future generations [8,9,15]. According to their fitness, individuals are selected for next generation. This is called selection, which ensures that the latter generations contain fitter individuals. This selection mechanism may be deterministic or stochastic. Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. The performance of any selection procedure is guided by the biasing and it affects the time complexity [8, 9,15].

### Genetic Algorithm in Optimization of Electric Machines:

The problem of optimizing the parameters of electric machines from its start up transient is a specific subgroup of the very general class of system identification problems. GAs proved to be generally applicable to this type of problem even though there are many alternatives in the literature. The problem can be formulated as follows:

Consider the electric machine as a system described as a state space model of the form

$$\begin{aligned} \dot{X} &= f(X, U, \hat{U}, \theta, Z) \\ Y &= g(X, U, \hat{U}, \theta, Z) \end{aligned}$$

Where  $X$  is the state vector of dimension  $n$ ;  $U$  the input vector of dimension  $r$ ;  $Y$  is the output vector of dimension  $m$ ;  $\theta$  is the dependent parameter vector of dimension  $m$ ;  $Z$  the independent parameter vector of dimension  $w$ . Under steady - state condition

$$\begin{aligned} 0 &= f(X_0, U_0, 0, \theta, Z), \\ Y &= g(X_0, U_0, 0, \theta, Z) \end{aligned}$$

Where, above equation denotes a steady state value. Presumably,  $U_0$  and  $Y_0$  are measurable and known, as it will be shown later in this work,  $X_0$  and  $\theta$  are therefore dependent on  $Z$  according to the above equation this means that the dynamic equation is solvable after  $Z$  is estimated. More general, let  $Z$  be a parametric vector that includes also model structure parameters such as the order of dynamic model; let  $S$  be a set of admissible parameters, i.e.  $Z \in S$ , which guarantee that a chosen model equation has a solution; let the observation space be  $O$ . Then the parameter- to - output mapping is written as:

$$\Phi: S \rightarrow O$$

The major procedure of an optimization based parameter estimation method is to search the best parameter vector  $Z^*$  in the search space  $S$ , which minimizes an error function  $E$ ,

$$E^* = \text{minimize } Z=Z^*, Z \in S E(Z)$$

The error function  $E$  is usually taken as a nonnegative and monotonically increasing function of output error:

$$E = \int_{t_0}^T J(\|Y_m(t) - Y_c(t)\|) dt \text{ (Continuous time)}$$

Where,  $[t_0, T]$  is the observation interval;  $\|\cdot\|$  denotes a norm;  $J(e)$  is a monotonically increasing function;  $k$  denotes the  $k$ th time sample;  $N$  is the number of all samples;  $(\cdot)_m$  denotes the measured (or true) values and  $(\cdot)_c$  computed values. The most widely used forms of  $J(e)$  are the square function, or absolute function, the square root function or their combinations. If an average model is required to fit a series of tests, the error function may be taken as a sum of all the test errors.

Better parameters generally result in less error function. In GAs, larger fitness would reproduce more offspring. This will most likely lead to better parameter estimation. Noting that the error function is always positive, fitness  $f$  is usually chosen as an inverse of the error function, so searching the minimum error function is equivalent to searching maximum fitness function:

$$F(Z) = 1/E(Z), \text{ maximize } f(Z) z=z^*, z \in S \leftrightarrow \text{ minimize } E(Z) z=z^*, z \in S$$

It will be shown later that by using this approach to identify parameters, the error between the response of an electric model with the actual parameters and GAs identified parameters can be minimized very well. This can be proved by simulating the output responses with both measured and estimated parameters.

## V. RESULTS

Initially an open loop IM speed response without any load has been given after the load has been placed on the IM, has shown in the fig1, a step input to the PI and PID controller has been given. As mentioned, initially the values of proportional, integral and derivative gain has been calculated using, trial and error and Z-N method and the system has been simulated and Fig 2 and fig3 shows the result with this initial parameters settings with and without load.

Then GA and have been implemented, and the optimized result with GA have implemented and Fig4 and fig 5 is the complete response of comparison of open loop and ZN and evolution algorithm (GA) without and with load, respectively.

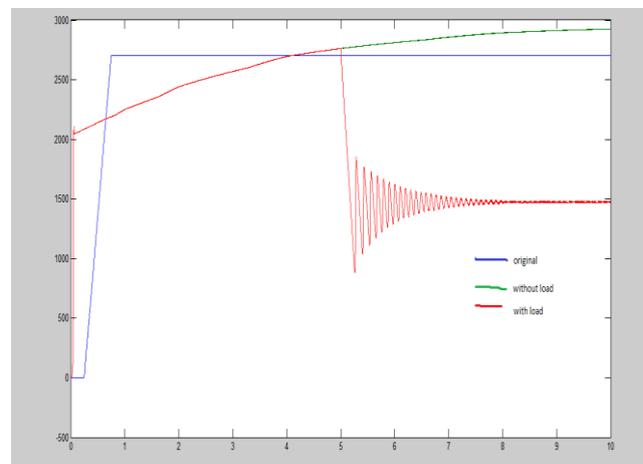


Fig1: open loop control of IM speed control with and without load

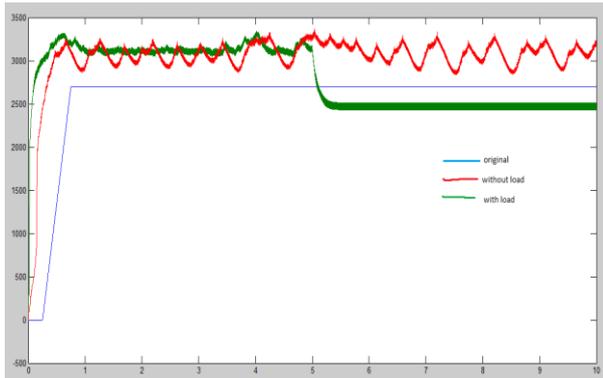


Fig 2: IM speed control by trail and error method with and without load

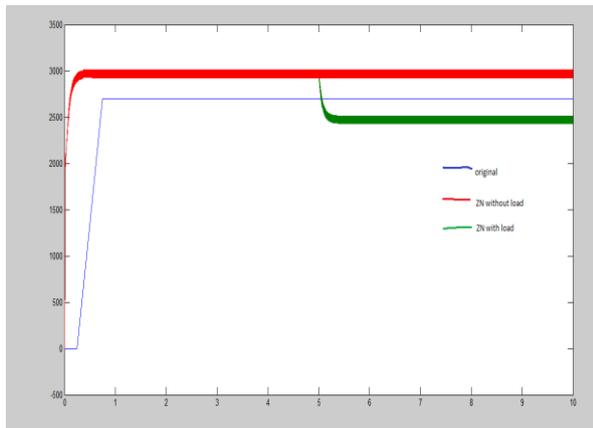


Fig 3: speed control of IM using ZN method with and without load

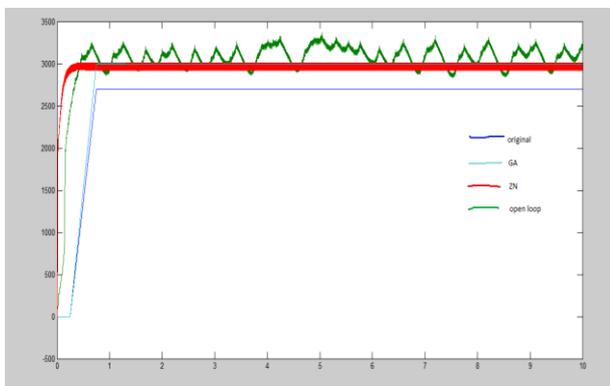


Fig 4: the performance speed control of IM without load disturbance for ZN and GA

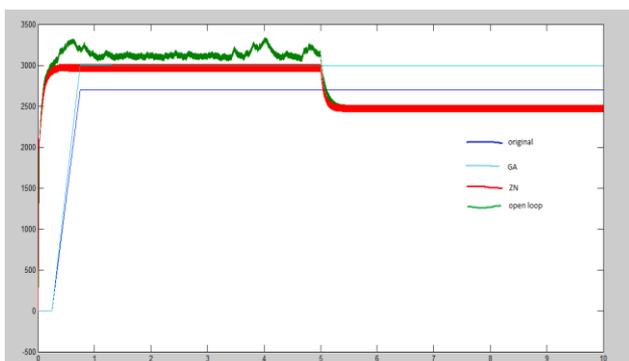


Fig 5: the performance speed control of IM with load disturbance for ZN and GA

## VI. CONCLUSION

Till now many authors have implemented the speed tracking problems in various methods. Thus it can be concluded that, by using the evolutionary algorithms(GA), the speed variation is better under load and no load condition , hence it provide much better speed response under no load and load condition, than that of conventional Z-N method .

## REFERENCES

- [1] Bose,B.K., "Modern Power Electronics and AC Drives", (Pearson Education Pte. Ltd ,First Indian reprint 2002,third reprint 2003)
- [2] Blaschke ,F., "The Principle of Field Orientation as applied to the new transvector closed loop control system for rotation field machine", Siemens Rev 1972,200-217
- [3] "Field Orientated Control of 3-PhaseAC-Motors", Texas Instruments Europe , Literature Number: BPRA073 February 1998
- [4] Casadei, D., Profumo , F. , Serra, G., Tani,A., "FOC and DTC: two viable schemes for induction motors torque control", IEEE trans. Power Electron, 17(September) (2002),779-787.
- [5] Novotonyand, R., Tipo,T., "Introduction to field orientation and high performance ac drives", IEEE Ind , Applicat.Soc.Ann Meeting , Tutorial Course1986.
- [6] Ogata,Katshuhiko., "Modern Control Engineering" (PHI Learning Private Limited ,Fifth edition,2010)Ang ,K., Chong, G., Li,Y. , "PID control system analysis, design, and technology," IEEE Trans.Control System Technology, vol. 13, pp. 559- 576, July 2005
- [8] Banerjee, T., Choudhury, S., Bera, J.N., "Offline optimization of PI/PID controller for a Vector Controlled Induction Motor Drive using Genetic Algorithm.-International Conference on "Electrical Power and Energy Systems" ICEPES 2010, 26-28<sup>th</sup> August 2010. On Pages192-195.
- [9] Krohling, R. A., Jaschek,H., Rey, J. P., "Designing PI/PID controller for a motion control system based on genetic algorithm", in Proc. 12th IEEE Int. Symp. Intell. Contr., Istanbul, Turkey, July 1997, pp. 125-130.
- [10] Eberhart,R., Kennedy,J., "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, vol. IV, Perth, Australia, 1995, pp. 1942-1948.
- [11] Mehdi, Nasri., Hossein Nezamabadi-pour, and Malihe Maghfoori, "A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor", in World Academy of Science, Engineering and Technology 26 2007
- [12] Gaing, Z.-L., "A particle swarm optimization approach for optimum design of PID controller in AVR system," IEEE Trans. Energy Conversion, vol. 19, pp. 384-391, June 2004
- [13] Particle Swarm Optimisation, <http://www.swarmintellig ence/tutori al.php>
- [14] Banerjee, T., Choudhury, S., Bera, J.N, Maity,A., "Off-line optimization of PI and PID controller for a vector controlled induction motor drive using PSO", (ICECE), 2010 International Conference on Electrical and Computer Engineering, 18-20 Dec. 2010, On Page(s): 74 – 77, ISBN: 978-1-4244-6277-3
- [15] Mahapatra, T.S,Roy, R., Mishra, S., Saha, N., Banerjee, T., Das, P., "Offline Optimisation of PI/PID Controller of a DC Shunt Machine Drive using Genetic Algorithm", ICNEAC 2011, July 8-10 2011, Narsapur, <http://www.iacqer.com>.
- [16] "Parameter estimation of an induction machine using advanced particle swarm optimization algorithms", IET Elect. Power Appl , November 2010, Volume 4, Issue 9, p.748-760
- [17] Panda, Siddhartha., Padhy, "Comparison of particle swarm optimization and genetic algorithm for TCSC-based controller design", Department of Electrical Engineering, Indian Institute of Technology, Roorkee, Uttaranchal 247667, India, International journal of Electrical and Electronics Engineering 1:5 2007. <http://www.waset.org/journal/ijeee/v1/v1-5-46.pdf>
- [18] K. B. Oldham and J. Spanier, The Fractional Calculus, Academic Press, New York, 1974.
- [19] I. Podlubny, "Fractional-order systems and PI $\lambda$ D $\mu$  controllers", IEEETrans. on Automatic Control, vol. 44, no. 1, pp. 208 - 213, 1999.
- [20] Y. Q. Chen, D. Xue, and H. Dou, "Fractional calculus and biomimeticcontrol", Proc. of First IEEE Int. Conf. on Robotics and Biomimetics pp. 901 - 904, 2004.
- [21] M. Nakagawa and K. Sorimachi, "Basic characteristics of a fractancedevice", IEICE Trans. Fundamentals, vol. 6, no. 12, pp. 1814-1818,1998.

- [22] J.Y. Cao and B. G. Cao, "Design of Fractional Order Controller Based on Particle Swarm Optimization", *International Journal of Control, Automation, and Systems*, vol. 4, no. 6, pp. 775-781, 2006.
- [23] A. Abraham, A. Biswas, S. Das, and S. Dasgupta, "Design of Fractional Order PID $\mu$  Controllers with an Improved Differential Evolution," *Genetic and Evolutionary Computation Conference, GECCO 2008*, ACM Press, ISBN: 978-1-60558-130-9, pp. 1445-1452, 2008.
- [24] B. M. Vinagre, I. Podlubny, L. Dorcak, and V. Feliu, "On fractional PID controllers: A frequency domain approach", *Proc. of IFAC Workshop on Digital Control – Past, Present and Future of PID Control*, pp. 53-58, 2000.
- [25] I. Petras, "The fractional order controllers: methods for their synthesis and application", *Journal of Electrical Engineering*, vol. 50, no. 9-10, pp. 284-288, 1999.
- [26] L. Dorcak, I. Petras, I. Kostial, and J. Terpak, "State-space controller design for the fractional-order regulated system", *Proc. of Int. Carpathian Control Conf.*, pp. 15-20, 2001.
- [27] Ma Chengbin and Y. Hori, "Fractional order control and its application of fractional order PID controller for robust two-inertia speed control" *Proc. of Fourth Int Power Electronics and Motion Control Conf.*, pp. 1477-1482, 2004.
- [28] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", *Proc. of IEEE Int. Conf. on Neural Networks*, pp. 1942-1948, 1995.
- [29] M. P. Song and G. H. Gu, "Research on particle swarm optimization: a review", *Proc. of Third Int Conf. On Machine Learning and Cybernetics*, pp. 2236-2241, 2004.
- [30] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer", *Proc. of IEEE Congress on Evolutionary Computation*, pp. 69-73, 1998.